

**I N T H E U N I T E D S T A T E S
P A T E N T A N D T R A D E M A R K O F F I C E**

Patent Application

Inventors: Sachin Garg et al.

Serial No.: 10/662724

Conf. No.: 1503

Filing Date: 9/15/2003

Art Unit: 2143

Examiner: Sikri, Anish

Docket No.: 630-044US

Title: Congestion Management in Telecommunications Networks

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

APPEAL BRIEF UNDER 37 CFR 41.67

Pursuant to 37 CFR 41.67, this brief is filed in support of the appeal in this application.

TABLE OF CONTENTS

| | |
|---|----|
| REAL PARTY IN INTEREST | 4 |
| RELATED APPEALS AND INTERFERENCES | 5 |
| STATUS OF CLAIMS..... | 6 |
| STATUS OF AMENDMENTS | 7 |
| SUMMARY OF THE CLAIMED SUBJECT MATTER..... | 8 |
| GROUNDS OF OBJECTION AND REJECTION TO BE REVIEWED ON APPEAL | |
| Ground 1: 35 U.S.C. 102 Rejection of Claims 1-12..... | 15 |
| ARGUMENTS | |
| Ground 1: 35 U.S.C. 102 Rejection of Claims 1-12..... | 16 |
| CONCLUSION | 23 |
| CLAIMS APPENDIX | 24 |
| EVIDENCE APPENDIX | 27 |
| RELATED PROCEEDINGS APPENDIX..... | 28 |

REAL PARTY IN INTEREST

The real party of interest in this application is the assignee of this application: Avaya Technology Corp., of Basking Ridge, NJ.

RELATED APPEALS AND INTERFERENCES

U.S. patent application Serial No. 10/662,728, filed 09/15/2003 (Attorney Docket: 630-045us) is related to this application. An appeal in that case is currently pending and awaiting review.

STATUS OF CLAIMS

Claims 1-12 stand rejected and are being appealed.

STATUS OF AMENDMENTS

All amendments have been entered.

SUMMARY OF THE CLAIMED SUBJECT MATTER

A typical computer network comprises a plurality of nodes and links and is depicted in Figure 1.

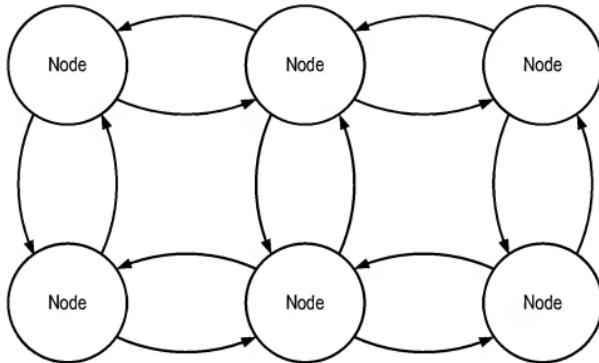


Figure 1 – A Typical Computer Network

Each link carries information from one node to another and can be, for example, an electrical cable, an optical cable, or a wireless relay. Each node switches information from one link to another and can be, for example, a switch, a router, or an access point.

Applicant's Specification [0008]

Most computer networks transmit information in discrete chunks called "protocol data units." Frames, packets, and datagrams are typical protocol data units. Applicant's Specification [0003] Protocol data units are injected into a network at one node and are passed from node to node, in bucket-brigade fashion, until they arrive at their destination. This is illustrated in Figure 2. Applicant's Specification [0002]

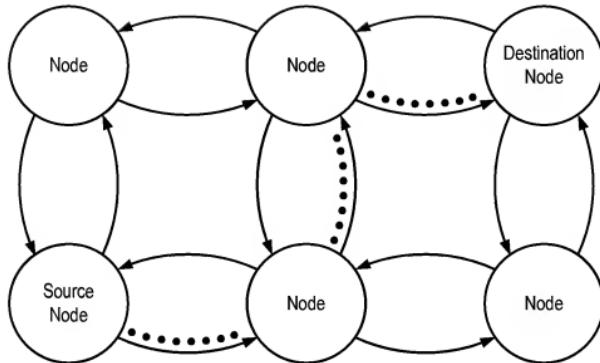


Figure 2 – Protocol Data Units Being Passed Bucket-Brigade Fashion

In some cases, a protocol data unit might spend a relatively short amount of time in a node before it is processed and transmitted on an output link. In other cases, a protocol data unit might spend a relatively long time. Applicant's Specification [0004]

One reason why a protocol data unit might spend a long time in a network node is because the output link on which the protocol data unit is to be transmitted is temporarily unavailable. Another reason why a protocol data unit might spend a long time in a network node is because a large number of protocol data units arrive at the node faster than the node can process and output them. Applicant's Specification [0005]

To accommodate this, a network node typically stores or "queues" a protocol data unit until it is transmitted. Sometimes, the protocol data units are stored in an "input queue" and sometimes the protocol data units are stored in an "output queue." An input queue might be employed when protocol data units arrive at the network node (in the short run) more quickly than they can be processed. An output queue might be employed when protocol data units arrive and are processed (in the short run) more quickly than they can be transmitted on the output link. Applicant's Specification [0006]

A queue has a finite capacity, and, therefore, it can fill up with protocol data units. When a queue is filled, the attempted addition of protocol data units to the queue causes

the queue to "overflow" with the result that the newly arrived protocol data units are discarded or "dropped." Dropped protocol units are lost forever and do not leave the network node. Applicant's Specification [0007]

A network node that comprises a queue that is dropping protocol data units is called "congested." For the purposes of this specification, a "**congestible node**" is defined as a network node (e.g. a switch, router, access point, *etc.*) that is susceptible to dropping protocol data units. Applicant's Specification [0008]

The loss of a protocol data unit is detrimental to the user of the protocol data unit, but the loss of any one protocol data unit does not have the same degree of impact as every other protocol data unit. In other words, the loss of some protocol data units is more injurious than the loss of some others. Applicant's Specification [0009]

When a congestible node is congested, or close to becoming congested, it can be prudent for the node to intentionally and proactively drop one or more protocol data units whose loss will be less consequential than to allow arriving protocol data units to overflow and be dropped and whose loss might be more consequential. To accomplish this, the node can employ an intelligent congestion management algorithm to decide:

- which protocol data units to drop,
- how many protocol data units to drop, and
- when to drop those protocol data units,

in order to:

- reduce injury to the affected communications, and
- lessen the likelihood of congestion in the congestible node.

Applicant's Specification [0010] How intelligent congestion management algorithms are designed is well known in the prior art and is not germane to the present invention.

Before intelligent congestion management algorithms were invented, there were thousands of congestible nodes in use. Today — after the development of intelligent congestion management algorithms — there are still thousands of congestible nodes in use. Why? Because intelligent congestion management algorithms cannot be retrofitted into most congestible nodes.

The solution to the problem of the existence of thousands of old congestible nodes without intelligent congestion management is — at least according to the prior art — to replace the old nodes with new nodes that do incorporate intelligent congestion management. The inventors of the present invention recognized that this answer is prohibitively expensive and in many cases suggests the replacement of nodes that are otherwise perfectly good. Clearly, a better solution was needed than to swap out thousands of otherwise perfectly-good nodes.

Furthermore, the proliferation of WiFi, Bluetooth, and Zigbee networks is fueling the need for inexpensive “lightweight” nodes that do not have the computing power to implement intelligent congestion management at all.

It was in this context that the applicants invented a device that can:

1. perform intelligent congestion management for a congestible node that cannot perform it for itself, and
2. be added to a computer network without any reconfiguration of the network or to the congestible node.

This is accomplished by inserting a new device — called a “protocol-data-unit excisor” — into the link carrying protocol data units to a congestible node. This is illustrated in Figure 3.

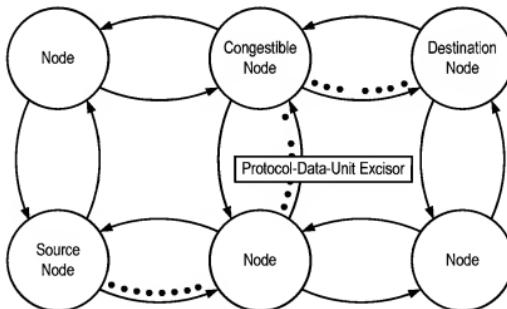


Figure 3 – Protocol-Data-Unit Excisor Inserted Into Link To Congestible Node

The protocol-data-unit excisor estimates whether there is congestion in the congestible node, and if so, discards some of the protocol data units before they get to the congestible node.¹ The net effect is the same as if the congestible node performed intelligent congestion management itself.

The protocol-data-unit excisor can estimate congestion in the congestible node either with or without the assistance of the congestible node.

When the protocol-data-unit excisor estimates congestion in the congestible node without the assistance of the congestible node, the protocol-data-unit excisor does so by analyzing the number and frequency of protocol data units *en route* to the congestible node and then making an educated guess. This is the subject of claims 7-12.

In contrast, the protocol-data-unit excisor can estimate congestion in the congestible node with the assistance of the congestible node. Some legacy nodes can transmit a signal back to a transmitting node telling the transmitting node to temporarily stop transmitting. The applicants recognized that this signal, and signals similar to it, could give the protocol-data-unit excisor a hint about whether there is congestion or not in the queue in the congestible node. In this case, the protocol-data-unit excisor analyzes the number and frequency of protocol data units *en route* to the congestible node and the signals from the congestible node, and then makes an educated guess. This is the subject of claims 1-6.

Claim 1 recites:

1. A method comprising:

receiving a first plurality of protocol data units at a first input of a protocol-data-unit excisor, wherein all of the protocol data units received at said first input are *en route* to a first congestible node;

receiving at said protocol-data-unit excisor a metric of a queue in said first congestible node; and

selectively dropping, at said protocol-data-unit excisor, one or more of said first plurality of protocol data units based on said metric of said queue in said first congestible node.

Claim 1 is supported in the specification at **Paragraphs [0015], [0052]-[0063], [0083]-[0088], and Figures 5 and 10.**

¹ The protocol-data-unit excisor can use any intelligent congestion management algorithm to decide which protocol data units to drop.

Independent claim 4 recites:

4. A protocol-data-unit excisor comprising:

a first input for receiving a first plurality of protocol data units, wherein all of the protocol data units received at said first input are en route to a first congestible node;

a second input for receiving a metric of a queue in said first congestible node; and

a processor for selectively dropping one or more of said first plurality of protocol data units based on said metric of said queue in said first congestible node.

Claim 4 is supported in the specification at **Paragraphs [0053]-[0051] and Box 302 in Figure 3 and all of Figure 4.**

Independent claim 7 recites:

7. A method comprising:

receiving a first plurality of protocol data units at a first input of a protocol-data-unit excisor, wherein all of the protocol data units received at said first input are en route to a first congestible node;

estimating in said protocol-data-unit excisor a first metric of a first queue of protocol data units in said first congestible node based on said first plurality of protocol data units; and

selectively dropping, at said protocol-data-unit excisor, one or more of said first plurality of protocol data units en route to said first congestible node based on said first metric.

Claim 7 is supported in the specification at **Paragraphs [0015], [0052]-[0063], [0083]-[0088], and Figures 5 and 10.**

Independent claim 10 recites:

10. A protocol-data-unit excisor comprising:

a first input for receiving a first plurality of protocol data units, wherein all of the protocol data units received at said first input are en route to a first congestible node; and

a processor for estimating a first metric of a first queue of protocol data units in said first congestible node based on said first plurality of protocol data units, and for selectively dropping one or more of said first plurality of protocol data units en route to said first congestible node based on said first metric.

Claim 10 is supported in the specification at **Paragraphs [0074]-[0082] and Box 802 in Figure 8 and all of Figure 9.**

GROUNDS OF OBJECTION AND REJECTION TO BE REVIEWED ON APPEAL

Ground 1: 35 U.S.C. 102 Rejection of Claims 1-12

Claims 1-12 have been rejected under 35 U.S.C. 102(b) as anticipated by N.A. Lyon et al., U.S. Patent 6,333,917 B1 (hereinafter "Lyon"). The applicants respectfully traverse the rejection.

ARGUMENTS**Ground 1: 35 U.S.C. 102 Rejection of Claims 1-12**

Claims 1-12 have been rejected under 35 U.S.C. 102(b) as anticipated by N.A. Lyon et al., U.S. Patent 6,333,917 B1 (hereinafter "Lyon"). The applicants respectfully traverse the rejection.

Claim 1 recites:

1. A method comprising:

receiving a first plurality of protocol data units at ***a first input*** of a protocol-data-unit excisor, ***wherein all of the protocol data units received at said first input are en route to a first congestible node;***

receiving at said protocol-data-unit excisor a metric of a queue in said first congestible node; and

selectively dropping, at said protocol-data-unit excisor, one or more of said first plurality of protocol data units based on said metric of said queue in said first congestible node.

(emphasis supplied)

Several of the salient limitations in the claims have been simply ignored by the Office in sustaining the rejection, or glossed over, and the applicants will highlight them and explain their significance to help the Board see the deficiencies in the reference.

The theory of invention embodied in this claim is that the protocol-data-unit excisor is performing intelligent congestion management ***as a proxy*** for another node — the congestible node. The limitations that embody this are:

First, the claim recites receiving a plurality of protocol data units ***at first input*** of a ***protocol-data-unit excisor, wherein all of the protocol data units are en route to a first congestible node.*** This limitation clarifies the fact that the protocol-data-excisor is a different device than the congestible node, and clarifies that all of the protocol data units arriving at one input of the protocol-data-unit excisor are going to the congestible node. The purpose of the latter is to prohibit switching on the part of the protocol-data-unit excisor, which excludes all switches that perform intelligent congestion management from the scope of the claim.

The Office cites the following portions of Lyon as anticipating the first step of the claim "receiving a first plurality of protocol data units at **a first input** of a protocol-data-unit excisor, **wherein all of the protocol data units received at said first input are en route to a first congestible node.**"

An apparatus consistent with the present invention is for use in controlling congestion in a network, the network including a switch having a switch fabric and a switch line card. The switch fabric has a queue for receiving packets transmitted over the network from a plurality of source connections, and the switch line card has a queue for receiving packets from the switch fabric. The apparatus comprises means for marking a first packet as an indication of congestion as the packet leaves the switch fabric queue, means for receiving the first packet at the switch line card queue, and means for marking a second packet in the switch line card queue in response to receiving the first packet.

Another apparatus consistent with the present invention is for use in controlling congestion across a link in a network, the network including a switch having a queue for receiving packets transmitted over the network from a plurality of source connections. The apparatus comprises means for marking a packet as an indication of congestion, means for removing the mark from the packet, and means for transmitting the mark across the link.

Yet another apparatus consistent with the present invention is for use in controlling congestion in a network, the network including a switch having a queue for receiving packets transmitted over the network from a plurality of source connections. The apparatus comprises means for determining a packet marking period defining how often the packets should be marked to indicate congestion, and means for determining in response to the packet marking period which of the packets should be marked.

Another apparatus consistent with the present invention is for use in controlling congestion in a network, the network including a switch having a queue for receiving packets transmitted over the network from a plurality of source connections. The apparatus comprises means for determining a single desired queue fill, means for determining a measure of the queue fill buffer occupancy based on a current queue fill and the single desired queue fill, and means for determining a packet marking period based on the measure of the queue fill buffer occupancy.

Yet another apparatus consistent with the present invention is for use in controlling congestion in a network, the network including a switch having a queue for receiving cells transmitted over the network from a plurality of source connections. The apparatus comprises means for determining a packet marking period based on a function that linearizes the response of the source connections to a packet being marked, and means for marking at least one packet according to the packet marking period.

A communications network consistent with the present invention comprises a plurality of sources transmitting packets to a plurality of destinations, and a plurality of packet switches, each packet switch

comprising a queue for receiving the packets transmitted over the network from the sources, and means for controlling congestion in the network including means for determining whether to mark a packet as an indication of congestion based on fill of the queue and means for marking the packet as the packet leaves the queue.

* * *

FIG. 4 depicts, in block diagram form, a RED+ system consistent with the present invention. The system, shown generally by reference numeral 44, operates within a switch and includes switch fabric 46 in communication with switch line card 48. At least one source 50 transmits packets as an "offered load" to switch fabric 46 over link 52. As shown, switch fabric 46 and line card 48 include buffer or queue 54 and 56, drop/tag section 58 and 60, and at least one RED+ engine 62 and 64, respectively. As explained in greater detail below, the RED+ engines monitor the fill of buffers 54 and 56. The packets are forwarded at a "service bandwidth" from line card 48 over link 66 to at least one destination 68, which is in communication with sources 50 through feedback path 70.

* * *

FIG. 5 illustrates one of the RED+ engines 62/64 shown in FIG. 4. Each RED+ engine includes marking (i.e., drop/tag) rate generator 74 and marking (i.e., drop/tag) decision generator 76. Marking rate generator 74 functions to observe the queue dynamics and determine the frequency of dropping/tagging packets. The marking rate generator may, for example, operate on queue fill (or some other congestion measure) and determine the rate at which packets should be marked. This marking rate is fed to marking decision generator 76, whose function is to generate instantaneous, packet-by-packet decisions to mark that conform to the given rate input. The packet to be marked may be the current packet, one already queued, or one yet to arrive.

This separation of marking rate and marking decision functions is based on real-time criticality. Marking decision generator 76 operates in real time, providing a decision for each dequeue opportunity, whereas marking rate generator 74 runs as fast as possible so as not to contribute significantly to sampling delay or aliasing. This may be achieved with a sampling period as long as 1,000 packet times or longer, depending on the network scenario. Because running with a faster sampling rate places increased demands on the data path width within the marking rate generator, it is desirable to make the sampling rate provisionable to insure against having insufficient range in the data path width and in RED+ parameters.

Lyon, Col. 3, line 56 to Col. 4, line 47; Col. 6, line 36-49; Col. 8 line 60 to Col. 9 line 18.

The applicants respectfully submit that the cited portion of Lyon does not, in fact, anticipate the limitation "receiving a first plurality of protocol data units at a **first input** of a

protocol-data-unit excisor, *wherein all of the protocol data units received at said first input are en route to a first congestible node.*" What in Lyon maps to the recited protocol-data-unit excisor? What in Lyon maps to the recited congestible node? And where does it say that a plurality of protocol data units received at one input of the protocol data unit excisor are all en route to the congestible node? The applicants respectfully submit that there is no such mapping.

Second, the claim recites receiving a "metric of a queue" in the congestible node, which is a hint that the queue in the congestible node might be overflowing or close to overflowing. In the prior art, this indication is used by a node receiving protocol data units to stop a node sending it protocol data units for whatever reason. This is, in essence, a type of non-destructive congestion management in the sense that flow control helps keep the congestible node from overflowing, but it is not "intelligent" in the sense that it is not used to selectively drop protocol data units. When the flow control is turned off, the transmission of all protocol data units is halted, regardless of whether they are "important" or "unimportant."

The Office cites the following portions of Lyon as anticipating the second step of the claim *"receiving at said protocol-data-unit excisor a metric of a queue in said first congestible node."*

An apparatus consistent with the present invention is for use in controlling congestion in a network, the network including a switch having a switch fabric and a switch line card. The switch fabric has a queue for receiving packets transmitted over the network from a plurality of source connections, and the switch line card has a queue for receiving packets from the switch fabric. The apparatus comprises means for marking a first packet as an indication of congestion as the packet leaves the switch fabric queue, means for receiving the first packet at the switch line card queue, and means for marking a second packet in the switch line card queue in response to receiving the first packet.

* * *

As shown in FIG. 2, end systems or nodes A, B, A', and B' connect to the network and serve as the source and sink of network traffic. Although unidirectional connections are shown and are implied by the orientation of queues 24 in switch 22, the connections shown, A to A' and B to B', are typically bidirectional, and the systems consistent with the present invention may operate in both directions. The mapping of traffic to queues 24 is arbitrary. That is, queue 24 may correspond to a single connection, a group

of connections, or all connections. Switch 22 need not be of any particular architecture and, for example, may be input-buffered, output-buffered, or a hybrid.

* * *

FIG. 11 presents a flowchart illustrating operation of connection selection consistent with the present invention. The connection selection function within the marking decision generator waits for a new packet to arrive (step 120). When a packet on a particular connection arrives, the connection selector updates connection metrics, e.g., a count of the number of packets for the connection that have arrived since the last time a packet on the connection was marked (step 122). There is at least one metric per connection. The connection selector also updates connection statistics (step 124), which are based on the connection metrics accumulated in step 122. There is at least one connection statistic per RED+ engine. If the connection selector receives a marking request from the marking request generator (step 126), it increments by one the number of pending marking requests (step 128) and proceeds to step 130. If there is no marking request, flow proceeds directly from step 126 to step 130. If there are no pending marking requests (step 130), flow proceeds to step 120, and the connection selector waits for the next packet. If the number of pending marking requests is greater than zero (step 130), then the connection selector determines whether the number exceeds the limit (step 132). If the number exceeds the limit, then flow proceeds directly to step 136. If the number does not exceed the limit, then the connection selector determines whether the connection metrics meet the statistical criteria for marking (step 134). If they do not, flow proceeds to step 120, and the connection selector waits for the next packet. If the statistical criteria are met in step 134, or if the number of pending requests exceeds the limit in step 132, then the connection selector decrements the number of pending requests by one (step 136) and resets the connection metrics (step 138). Finally, the connection selector marks the packet on the selected connection (step 140).

Lyon, Col. 3, line 56-67; Col. 6. line 7-19; Col. 14 line 54 to Col. 15 line 20.

The applicants respectfully submit that the cited portion of Lyon does not, in fact, anticipate the limitation "**receiving at said protocol-data-unit excisor a metric of a queue in said first congestible node.**" Following from the earlier argument, what in Lyon maps to the protocol-data-unit excisor? What maps to the congestible node? And where does Lyon say that an indication of congestion in the congestible node is received by the protocol-data-unit excisor? The applicants respectfully submit that there is no such mapping.

For these reasons, the applicants respectfully submit that the rejection of claim 1 is traversed.

Because claims 2 and 3 depend on claim 1, the applicants respectfully submit that the rejection of them is also traversed.

Claim 4 recites:

4. A protocol-data-unit excisor comprising:
a first input for receiving a first plurality of protocol data units, **wherein all of the protocol data units received at said first input are en route to a first congestible node;**
a second input for receiving a metric of a queue in said first congestible node; and
a processor for selectively dropping one or more of said first plurality of protocol data units based on said metric of said queue in said first congestible node.
(emphasis supplied)

For essentially the same reasons as those given with respect to claim 1, the applicants respectfully submit that the rejection of it is traversed.

Because claims 5 and 6 depend on claim 4, the applicants respectfully submit that the rejection of them is also traversed.

Claim 7 recites:

7. A method comprising:
receiving a first plurality of protocol data units at a first input of a protocol-data-unit excisor, **wherein all of the protocol data units received at said first input are en route to a first congestible node;**
estimating in said protocol-data-unit excisor a first metric of a first queue of protocol data units in said first congestible node based on said first plurality of protocol data units; and
selectively dropping, at said protocol-data-unit excisor, one or more of said first plurality of protocol data units en route to said first congestible node based on said first metric.
(emphasis supplied)

For essentially the same reasons as those given with respect to claim 1, the applicants respectfully submit that the rejection of it is traversed.

Because claims 8 and 9 depend on claim 7, the applicants respectfully submit that the rejection of them is also traversed.

Claim 10 recites:

10. A protocol-data-unit excisor comprising:

a first input for receiving a first plurality of protocol data units, **wherein all of the protocol data units received at said first input are en route to a first congestible node;** and

a processor for estimating a first metric of a first queue of protocol data units in said first congestible node based on said first plurality of protocol data units, and for selectively dropping one or more of said first plurality of protocol data units en route to said first congestible node based on said first metric.

(emphasis supplied)

For essentially the same reasons as those given with respect to claim 1, the applicants respectfully submit that the rejection of it is traversed.

Because claims 11 and 12 depend on claim 10, the applicants respectfully submit that the rejection of them is also traversed.

CONCLUSION

The applicants have demonstrated that the logic underlying the Office's rejection is untenable, and, therefore, that the rejection is not sustainable. For this reason, the applicants respectfully request the Board of Appeals to reverse the decision of the Examiner as provided for in 37 C.F.R. 41.50(a).

Respectfully,
Sachin Garg et al.

By **/Jason Paul DeMont/**
Jason Paul DeMont
Reg. No. 35,793
Attorney for Applicants
732-687-7990

DeMont & Breyer, L.L.C.
Suite 250
100 Commons Way
Holmdel, NJ 07733
United States of America

Claims Appendix**1. (previously presented)** A method comprising:

receiving a first plurality of protocol data units at a first input of a protocol-data-unit excisor, wherein all of the protocol data units received at said first input are en route to a first congestible node;

receiving at said protocol-data-unit excisor a metric of a queue in said first congestible node; and

selectively dropping, at said protocol-data-unit excisor, one or more of said first plurality of protocol data units based on said metric of said queue in said first congestible node.

2. (previously presented) The method of claim 1 wherein said protocol-data-unit excisor decides whether to drop a protocol data unit based on Random Early Detection.

3. (previously presented) The method of claim 1 further comprising:

receiving a second plurality of protocol data units at a second input of said protocol-data-unit excisor, wherein all of the protocol data units received at said second input are en route to a second congestible node;

receiving at said protocol-data-unit excisor a metric of a queue in said second congestible node; and

selectively dropping, at said protocol-data-unit excisor, one or more of said second plurality of protocol data units based on said metric of said queue in said second congestible node.

4. (previously presented) A protocol-data-unit excisor comprising:

a first input for receiving a first plurality of protocol data units, wherein all of the protocol data units received at said first input are en route to a first congestible node;

a second input for receiving a metric of a queue in said first congestible node; and

a processor for selectively dropping one or more of said first plurality of protocol data units based on said metric of said queue in said first congestible node.

5. (previously presented) The protocol-data-unit excisor of claim 4 wherein said protocol-data-unit excisor decides whether to drop a protocol data unit based on Random Early Detection.

6. (previously presented) The protocol-data-unit excisor of claim 4 further comprising:

a third input for receiving a second plurality of protocol data units, wherein all of the protocol data units received at said third input are en route to a second congestible node;

a fourth input for receiving a metric of a queue in said second congestible node;

wherein said processor is also for selectively dropping one or more of said second plurality of protocol data units based on said metric of said queue in said second congestible node.

7. (previously presented) A method comprising:

receiving a first plurality of protocol data units at a first input of a protocol-data-unit excisor, wherein all of the protocol data units received at said first input are en route to a first congestible node;

estimating in said protocol-data-unit excisor a first metric of a first queue of protocol data units in said first congestible node based on said first plurality of protocol data units; and

selectively dropping, at said protocol-data-unit excisor, one or more of said first plurality of protocol data units en route to said first congestible node based on said first metric.

8. (previously presented) The method of claim 7 wherein said protocol-data-unit excisor decides whether to drop a protocol data unit based on Random Early Detection.

9. (previously presented) The method of claim 7 further comprising:

receiving a second plurality of protocol data units at a second input of said protocol-data-unit excisor, wherein all of the protocol data units received at said second input are en route to a second congestible node;

estimating in said protocol-data-unit excisor a second metric of a second queue of protocol data units in said second congestible node based on said second plurality of protocol data units; and

selectively dropping, at said protocol-data-unit excisor, a one or more of said second plurality of protocol data units en route to said second congestible node based on said second metric.

10. (previously presented) A protocol-data-unit excisor comprising:

a first input for receiving a first plurality of protocol data units, wherein all of the protocol data units received at said first input are en route to a first congestible node; and
a processor for estimating a first metric of a first queue of protocol data units in said first congestible node based on said first plurality of protocol data units, and for selectively dropping one or more of said first plurality of protocol data units en route to said first congestible node based on said first metric.

11. (previously presented) The protocol-data-unit excisor of claim 10 wherein said processor for selectively dropping one or more protocol data units decides whether to drop a protocol data unit based on Random Early Detection.

12. (previously presented) The protocol-data-unit excisor of claim 10 further comprising:

a second input for receiving a second plurality of protocol data units, wherein all of the protocol data units received at said second input are en route to a second congestible node; and

a processor for estimating a second metric of a second queue of protocol data units in said second congestible node based on said second plurality of protocol data units, and for selectively dropping one or more of said second plurality of protocol data units en route to said second congestible node based on said second metric.

Evidence Appendix

There is no evidence submitted pursuant to 37 CFR §§ 1.130, 1.131, or 1.132.

Related Proceedings Appendix

U.S. patent application Serial No. 10/662,728, filed 09/15/2003 (Attorney Docket: 630-045us) is related to this application. An appeal in that case is currently pending and awaiting review.